# Barely_production_price

Fentaw Abitew

2022-11-18

## Part 1: Overview

This test is inspired by a recent EPIC research project. Your task is to perform basic data cleaning, prepare a final dataset for analysis, provide short answers to prompts, and create publication-quality figures and tables.

This task should take five hours or less, but you will have 48 hours total to complete it. The goal of the task is to give you an opportunity to demonstrate your coding competency and your conceptual understanding of empirical economics research. A perfect grade is not a prerequisite for consideration for the position. For the coding portion of this test, we will accept code in Python, R, or STATA. You may consult any pre-existing online programming resources, but you may not ask other people for help. If you find any of these instructions to be confusing, please proceed in a way that you find relevant and reasonable, and list your assumptions in your writeup. Once you have completed the sections below, please submit the following in a .zip file:

- Well-commented code in the language of your choice (.do files for Stata, .R or .Rmd files for R, .py or .ipynb files for Python, etc.),
- The final dataset from Section 2,
- The final graphs and tables from Section 3,
- A short document answering the questions from Sections 2-4

## Part 2: Data Cleaning

The central task of this section is to merge production data and price data into one dataset. Both data are sourced from the US Department of Agriculture. Both datasets contain annual data from 1990 to 2018 A brief introduction on the datasets:

- Barley production.csv lists the barley production in bushels by agricultural district. The agricultural district is an administrative division between the county and state levels.
- Barley price.csv lists the mean price received by farmers per bushel of barley by state. Ignore the distinction between the marketing year and the calendar year. We want the final dataset to be:
- a panel with three dimensions: year, agricultural district, state
- in each row it contains: barley production and price

```r
# required lib
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
```

```
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(magrittr)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:magrittr':
##
##     extract
```

```r
library(ggplot2)
library(knitr)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
getwd()
```

```
## [1] "/Users/fam/Desktop/Desktop - FENTAW’s MacBook Air/American_U/R_programming/U_chicago_epic"
```

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8     v stringr 1.4.1
## v purrr   0.3.5     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x purrr::set_names() masks magrittr::set_names()
```

```r
library(fs)
library(dplyr)
library(stringr)
```

```r
#combine the price .csv in one file
file_path<-"/Users/fam/Desktop/Desktop - FENTAW’s MacBook Air/American_U/R_programming/U_chicago_epic/ra
barely_price_all <- list.files(path = file_path,  # Identify all CSV files
                       pattern = "*.csv", full.names = TRUE) %>%
  lapply(read_csv) %>%                              # Store all files in list
  bind_rows                                         # Combine data sets into one data set Print data to
```

```
## Rows: 28 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 28 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 28 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 27 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 27 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 27 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 27 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 28 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 28 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 28 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 28 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 29 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 28 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 28 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 28 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 27 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 27 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 27 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 27 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 23 Columns: 21
## -- Column specification ---------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 23 Columns: 21
## -- Column specification ---------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 23 Columns: 21
## -- Column specification ---------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 23 Columns: 21
## -- Column specification ---------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 23 Columns: 21
## -- Column specification ---------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 23 Columns: 21
## -- Column specification ---------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 23 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 16 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 16 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 23 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): program, period, geolevel, state, commodity, dataitem, domain, doma...
## dbl (4): year, stateansi, watershed_code, value
## lgl (9): weekending, agdistrict, agdistrictcode, county, countyansi, zipcode...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
dim(barely_price_all)
```

```
## [1] 741  21
```

```r
#production file
file_path<-"/Users/fam/Desktop/Desktop - FENTAW’s MacBook Air/American_U/R_programming/U_chicago_epic/ra
barely_production_all <- list.files(path = file_path,pattern = "*.csv", full.names = TRUE) %>%
  lapply(read_csv) %>%
  bind_rows
```

```
## Rows: 169 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
```

```
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 170 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 171 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 167 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 166 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 153 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 156 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 151 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 155 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 147 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 134 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 138 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 138 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 134 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 133 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 128 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 129 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 124 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 115 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 72 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 74 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 73 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 68 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 66 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 65 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 65 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 52 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 47 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
```

```
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 45 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): program, period, geolevel, state, agdistrict, commodity, dataitem, ...
## dbl (4): year, stateansi, agdistrictcode, watershed_code
## num (1): value
## lgl (7): weekending, county, countyansi, zipcode, region, watershed, cv
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
view(barely_production_all)
view(barely_price_all)
```

```
#merge the production and price data to create single dataframe
barely_production_all %>%
  select(year, state, agdistrict, value) -> production_col_final
head(production_col_final)
```

```
## # A tibble: 6 x 4
##    year state      agdistrict      value
##   <dbl> <chr>      <chr>           <dbl>
## 1  1990 ARIZONA    CENTRAL       1331300
## 2  1990 ARIZONA    NORTHERN        10300
## 3  1990 ARIZONA    SOUTHEASTERN   107400
## 4  1990 ARIZONA    SOUTHWESTERN   126000
## 5  1990 CALIFORNIA CENTRAL COAST 1339000
## 6  1990 CALIFORNIA NORTHEAST     1450000
```

```
barely_price_all%>%
  select(year,state, value)->price_col_final
head(price_col_final)
```

```
## # A tibble: 6 x 3
##    year state      value
##   <dbl> <chr>      <dbl>
## 1  1990 ALASKA      3.3
## 2  1990 ARIZONA     2.79
## 3  1990 CALIFORNIA  2.62
## 4  1990 COLORADO    3.06
## 5  1990 DELAWARE    1.89
## 6  1990 IDAHO       2.62
```

```
barely_price_production <- merge(production_col_final,price_col_final, by= c("year", "state"))
head(barely_price_production)
```

```
##   year      state    agdistrict value.x value.y
## 1 1990    ARIZONA       CENTRAL 1331300    2.79
## 2 1990    ARIZONA      NORTHERN   10300    2.79
## 3 1990    ARIZONA  SOUTHEASTERN  107400    2.79
## 4 1990    ARIZONA  SOUTHWESTERN  126000    2.79
## 5 1990 CALIFORNIA CENTRAL COAST 1339000    2.62
## 6 1990 CALIFORNIA     NORTHEAST 1450000    2.62
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following object is masked from 'package:purrr':
##
##     transpose

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
# assigning new names to the columns of the data frame
barely_price_production %>%
  setnames(old=c("year","state","agdistrict","value.x","value.y"), new= c("year","state", "agricultural_
```

```
dim(barely_price_production_final)
```

```
## [1] 3405    5
```

```
barely_price_production_final%>%
  discard(is.null) ->barely_price_production_final
```

# Part 3: Data Exploration

The data exploration in this section would be conducted at state-year level.

## 3.1 Time Series Plot: Price

For each year, compute the weighted average of price over all states, where each state's weight is its production in bushels in that year. Then, plot this weighted average over the time period from 1990 to 2018.

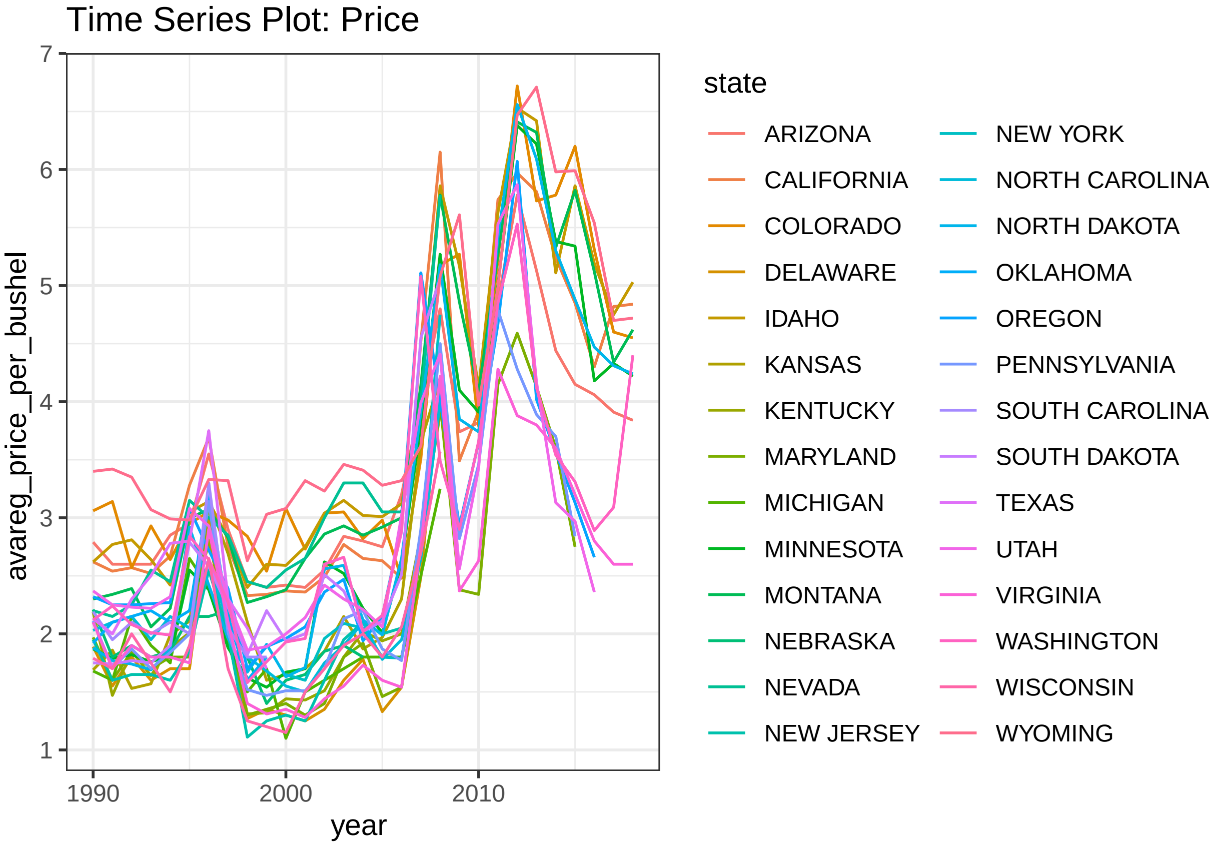
```
# Note: I read the question as : For each year, compute the weighted average of price over all states,

barely_price_production_final %>%
  group_by(state,year) %>%
  summarise(avarege_price_per_bushel = mean(price_per_bushel)) -> mean_price_state
```

```
## `summarise()` has grouped output by 'state'. You can override using the
## `.groups` argument.
```

```
head(mean_price_state)
```

```
## # A tibble: 6 x 3
## # Groups:   state [1]
##   state    year avarege_price_per_bushel
##   <chr>   <dbl>                    <dbl>
## 1 ARIZONA  1990                     2.79
## 2 ARIZONA  1991                     2.60
## 3 ARIZONA  1992                     2.60
## 4 ARIZONA  1993                     2.60
## 5 ARIZONA  1994                     2.85
## 6 ARIZONA  1995                     2.95
```

```
library(ggplot2)
ggplot(data = mean_price_state, mapping = aes(y = avarege_price_per_bushel, x = year, color=
state)) +
xlab("year ")+
ylab("avareg_price_per_bushel ")+
ggtitle("Time Series Plot: Price")+
theme_bw()+
geom_line()
```



## 3.2 Time Series Plot: Production

Find the top 3 states in terms of barley production in 2018. Plot the time series of production for these 3 states in the same plot, over the period from 1990 to 2018. Scale the production variable so that it is in millions of bushels.

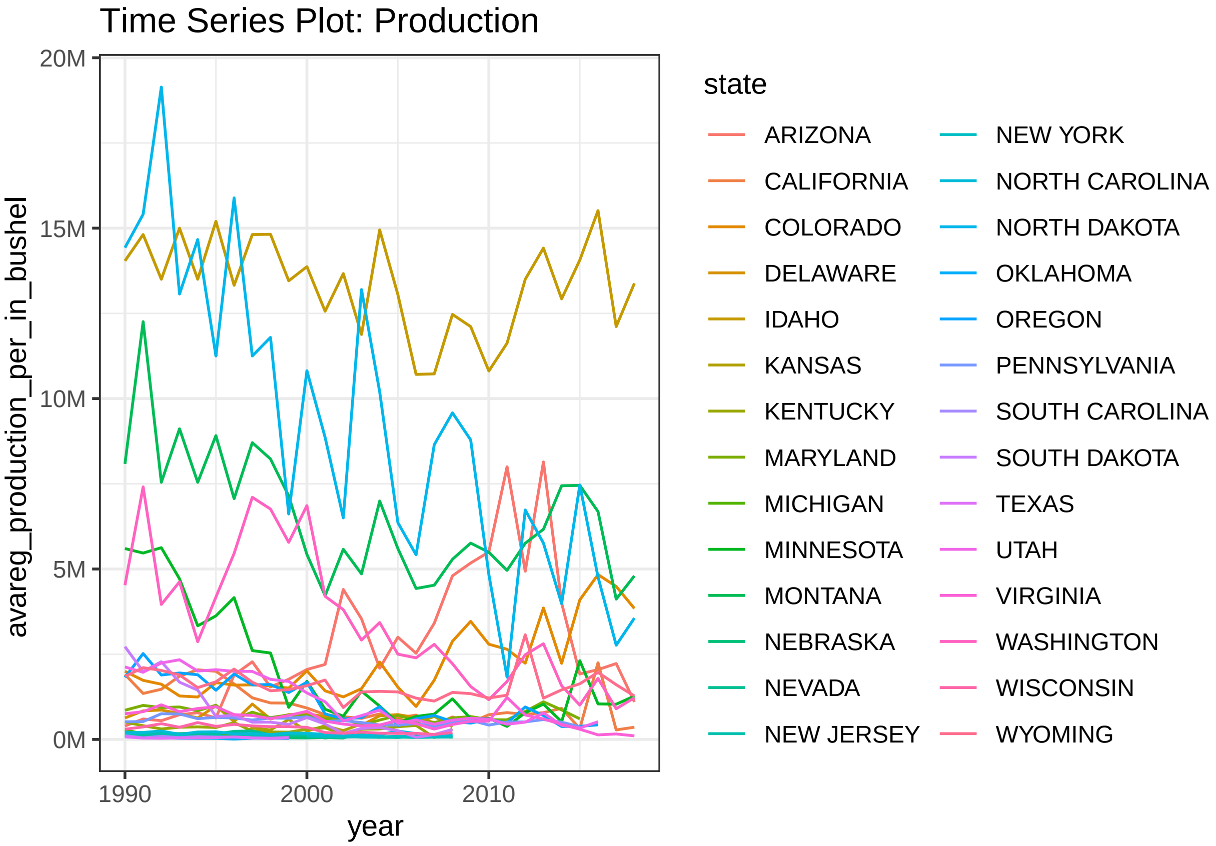
```
# Avaraged yearly production by state
barely_price_production_final%>%
  group_by(state,year) %>%
  summarise(avarege_production_in_bushel = mean(production_in_bushel)) -> mean_production_state
```

```
## `summarise()` has grouped output by 'state'. You can override using the
## `.groups` argument.
```

```
# Filter 2018 year production
mean_production_state %>%
  filter(year==2018) -> production_18
# arrange desc order to get top 3 producer
head(arrange(production_18, desc(avarege_production_in_bushel)),3)
```

```
## # A tibble: 3 x 3
## # Groups:   state [3]
##   state     year avarege_production_in_bushel
##   <chr>    <dbl>                        <dbl>
## 1 IDAHO     2018                     13382500
## 2 MONTANA   2018                      4800000
## 3 COLORADO  2018                      3842500
```

```
library(ggplot2)
ggplot(data = mean_production_state, mapping = aes(y = avarege_production_in_bushel, x = year, color= st
scale_y_continuous(labels =scales::unit_format(suffix = "M", scale = 1e-6))+
xlab("year ")+
ylab("avareg_production_per_in_bushel ")+
ggtitle("Time Series Plot: Production")+
theme_bw()+
geom_line()
```



## 3.3 Summary Table Create a summary table where the rows are specific states (Idaho, Minnesota, Montana,North Dakota, and Wyoming) and the columns are decades (1990-1999, 2000-2009, and 2010-2018). The elements of the table are mean annual state-level production, by decade and state. Scale the production variable so that it is in millions of bushels.

```
head(barely_price_production_final)
```

```
##   year      state agricultural_district production_in_bushel price_per_bushel
## 1 1990    ARIZONA               CENTRAL              1331300             2.79
## 2 1990    ARIZONA              NORTHERN                10300             2.79
## 3 1990    ARIZONA          SOUTHEASTERN               107400             2.79
## 4 1990    ARIZONA          SOUTHWESTERN               126000             2.79
## 5 1990 CALIFORNIA         CENTRAL COAST              1339000             2.62
## 6 1990 CALIFORNIA             NORTHEAST              1450000             2.62
```

```
barely_price_production_final%>%
  filter(state==c("IDAHO", "MINNESOTA", "MONTANA", "NORTH DAKOTA", "WYOMING"))%>%
  group_by(state, year) %>%
  summarise(avarege_production_in_bushel = mean(production_in_bushel))->sum_table
```

```
## `summarise()` has grouped output by 'state'. You can override using the
## `.groups` argument.
```

```
sum_table%>%
  mutate(year_bin = cut(year, breaks = c(1990,1999,2009,2018),dig.lab=4, labels = c("1990-1999","2000-20
#c("1990-1999","2000-2009","2010-2018")
```

```
head(sum_tablec)
```

```
## # A tibble: 6 x 4
## # Groups:   state [1]
##   state  year avarege_production_in_bushel year_bin
##   <chr> <dbl>                        <dbl> <fct>
## 1 IDAHO  1990                     31180000 <NA>
## 2 IDAHO  1991                      8833000 1990-1999
## 3 IDAHO  1992                     29850000 1990-1999
## 4 IDAHO  1993                     30246000 1990-1999
## 5 IDAHO  1994                      3500000 1990-1999
## 6 IDAHO  1995                      2839000 1990-1999
```

```
sum_tablec%>%
  group_by(state,year_bin, avarege_production_in_bushel) %>%
  summarise(avarege_production_in_bushel=mean(avarege_production_in_bushel*1e-06)) ->final_df
```

```
## `summarise()` has grouped output by 'state', 'year_bin'. You can override using
## the `.groups` argument.
```

```
head(final_df)
```

```
## # A tibble: 6 x 3
## # Groups:   state, year_bin [1]
##   state year_bin  avarege_production_in_bushel
##   <chr> <fct>                            <dbl>
## 1 IDAHO 1990-1999                         2.21
## 2 IDAHO 1990-1999                         2.84
## 3 IDAHO 1990-1999                         3.5
## 4 IDAHO 1990-1999                         8.83
## 5 IDAHO 1990-1999                        15.5
## 6 IDAHO 1990-1999                        29.8
```

```
 final_df %>%
  select(state, year_bin, avarege_production_in_bushel) %>%
  mutate(row = row_number()) %>%
  spread(year_bin, avarege_production_in_bushel)
```

```
## # A tibble: 48 x 6
## # Groups:   state [5]
##    state       row `1990-1999` `2000-2009` `2010-2018` `<NA>`
##    <chr>     <int>       <dbl>       <dbl>       <dbl>  <dbl>
##  1 IDAHO         1      2.21         0.97       0.43   31.2
##  2 IDAHO         2      2.84         2.7        0.53   NA
##  3 IDAHO         3      3.5          4.58       0.57   NA
##  4 IDAHO         4      8.83        14.0        2.27   NA
##  5 IDAHO         5     15.5         15.0        2.34   NA
##  6 IDAHO         6     29.8         15.8       19.0    NA
##  7 IDAHO         7     30.2         16.8       25.5    NA
##  8 IDAHO         8     31.1         29.9       NA      NA
##  9 IDAHO         9     NA           31.5       NA      NA
## 10 MINNESOTA     1      0.0976       0.104      0.0717  0.299
## # ... with 38 more rows
```

```
#head(summary_table,10)
```

# 4 Short Answer

Our goal is to estimate the sensitivity of US farmers' barley production to barley price, using the provided data, at the level of agricultural district by year.

```
barely_price_production_final%>%
  group_by(agricultural_district,year,price_per_bushel, state) %>%
  summarise(avarege_production_in_bushel = mean(production_in_bushel)) -> agricultural_district_mean_pr
```

```
## `summarise()` has grouped output by 'agricultural_district', 'year',
## 'price_per_bushel'. You can override using the `.groups` argument.
```

```
head(agricultural_district_mean_production)
```

```
## # A tibble: 6 x 5
## # Groups:   agricultural_district, year, price_per_bushel [6]
##   agricultural_district  year price_per_bushel state avarege_production_in_bus~1
##   <chr>                 <dbl>            <dbl> <chr>                       <dbl>
## 1 BLACKLANDS             1990             2.13 TEXAS                       34000
## 2 BLACKLANDS             1991             2    TEXAS                       51000
## 3 BLACKLANDS             1992             2.3  TEXAS                       28100
## 4 BLACKLANDS             1993             2.5  TEXAS                       46000
## 5 BLACKLANDS             1994             2.78 TEXAS                       36000
## 6 BLACKLANDS             1995             2.8  TEXAS                       37000
## # ... with abbreviated variable name 1: avarege_production_in_bushel
```

- First write down a regression equation of a linear model of production on a constant and price. We want the coefficient on price to have the interpretation of an elasticity. Ensure that the terms are properly indexed. Report the results of this regression, and interpret the coefficient on price.

```
#production_hat= beta0_hat + beta1_hat(price)
reg<-lm(log(avarege_production_in_bushel)~log(price_per_bushel), data =agricultural_district_mean_produ
summary(reg)
```

```
##
## Call:
## lm(formula = log(avarege_production_in_bushel) ~ log(price_per_bushel),
##     data = agricultural_district_mean_production)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0651 -1.3869 -0.1377  1.3911  5.3691
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           11.75760    0.08313  141.43   <2e-16 ***
## log(price_per_bushel)  1.22441    0.08339   14.68   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.916 on 3403 degrees of freedom
## Multiple R-squared:  0.05958,    Adjusted R-squared:  0.05931
## F-statistic: 215.6 on 1 and 3403 DF,  p-value: < 2.2e-16
```

Interpretation: A 1% increase in barely price would lead to 1.2% increase of barely production.

- What variables do you think we should control for? Choose two and explain why they might help us identify the coefficient on price. These variables need not to be in the original dataset.

#Location(state or agriculture district) and production_year(may contain weather shock in it)

```
reg<-lm(log(avarege_production_in_bushel)~log(price_per_bushel) + year + agricultural_district, data = a
summary(reg)
```

```
##
## Call:
## lm(formula = log(avarege_production_in_bushel) ~ log(price_per_bushel) +
##     year + agricultural_district, data = agricultural_district_mean_production)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9979 -1.0801  0.0394  1.0711  4.8107
##
## Coefficients:
##                                                  Estimate Std. Error t value
## (Intercept)                                     43.360078  10.173561   4.262
## log(price_per_bushel)                            1.222029   0.101516  12.038
## year                                            -0.016855   0.005122  -3.291
## agricultural_districtCENTRAL                     2.518115   0.544480   4.625
## agricultural_districtCENTRAL COAST               2.724126   0.627981   4.338
## agricultural_districtCENTRAL COASTAL             0.313312   0.712968   0.439
## agricultural_districtCENTRAL PIEDMONT            2.508323   0.662668   3.785
## agricultural_districtCROSS TIMBERS              -0.767456   0.803460  -0.955
## agricultural_districtEAST                        4.582513   0.606221   7.559
## agricultural_districtEAST CENTRAL                2.416005   0.551294   4.382
## agricultural_districtEASTERN                     2.033146   0.575634   3.532
## agricultural_districtEASTERN OR MOUNTAIN         0.846633   1.778143   0.476
## agricultural_districtEDWARDS PLATEAU            -0.765829   0.757499  -1.011
## agricultural_districtLOWER EASTERN SHORE         2.460885   0.632177   3.893
## agricultural_districtMIDWESTERN                  2.797787   0.663071   4.219
## agricultural_districtNORTH                       3.136318   0.602666   5.204
## agricultural_districtNORTH CENTRAL               3.121836   0.548949   5.687
## agricultural_districtNORTHEAST                   2.532542   0.545718   4.641
## agricultural_districtNORTHERN                    2.124199   0.566201   3.752
## agricultural_districtNORTHERN COAST             -0.994266   1.312482  -0.758
## agricultural_districtNORTHERN COASTAL            0.122041   0.712968   0.171
## agricultural_districtNORTHERN HIGH PLAINS        0.862907   0.757499   1.139
## agricultural_districtNORTHERN LOW PLAINS        -1.367583   1.115093  -1.226
## agricultural_districtNORTHERN MOUNTAIN           1.391206   0.662668   2.099
## agricultural_districtNORTHERN PIEDMONT           1.332053   0.662668   2.010
## agricultural_districtNORTHWEST                   2.872233   0.547635   5.245
## agricultural_districtNORTHWEST AND MOUNTAIN         0.113242   0.683001   0.166
## agricultural_districtNORTHWESTERN                   0.584333   0.683672   0.855
## agricultural_districtOTHER DISTRICTS, ALL COUNTIES  0.591386   0.548503   1.078
## agricultural_districtPANHANDLE                      0.317376   0.757595   0.419
## agricultural_districtSACRAMENTO VALLEY              2.171769   0.625027   3.475
## agricultural_districtSAN JOAQUIN VALLEY             3.627315   0.638203   5.684
## agricultural_districtSAN LUIS VALLEY                4.587878   0.628045   7.305
## agricultural_districtSIERRA MOUNTAINS              -1.086626   0.757565  -1.434
## agricultural_districtSISKIYOU-SHASTA                2.541800   0.634502   4.006
## agricultural_districtSOUTH                          0.450404   0.612112   0.736
## agricultural_districtSOUTH CENTRAL                  2.596990   0.549076   4.730
## agricultural_districtSOUTHEAST                      2.207205   0.547010   4.035
## agricultural_districtSOUTHEASTERN                   2.453624   0.582538   4.212
## agricultural_districtSOUTHERN                       1.740451   0.560374   3.106
## agricultural_districtSOUTHERN CALIFORNIA            1.264896   0.651162   1.943
## agricultural_districtSOUTHERN COASTAL               0.546085   0.662668   0.824
## agricultural_districtSOUTHERN LOW PLAINS           -0.779365   0.757499  -1.029
## agricultural_districtSOUTHERN PIEDMONT              2.056103   0.662668   3.103
## agricultural_districtSOUTHWEST                      1.693376   0.547932   3.090
## agricultural_districtSOUTHWESTERN                   0.935953   0.588589   1.590
## agricultural_districtTRANS-PECOS                   -0.501552   1.002151  -0.500
## agricultural_districtUPPER EASTERN SHORE            3.724721   0.632177   5.892
## agricultural_districtUPPER PENINSULA                2.111526   0.663183   3.184
## agricultural_districtWEST                           2.113450   0.625304   3.380
## agricultural_districtWEST CENTRAL                   2.018372   0.553777   3.645
## agricultural_districtWESTERN                        1.143647   0.570395   2.005
## agricultural_districtWESTERN MOUNTAIN              -0.685839   0.675777  -1.015
##                                                    Pr(>|t|)
## (Intercept)                                        2.08e-05 ***
## log(price_per_bushel)                               < 2e-16 ***
## year                                               0.001010 **
## agricultural_districtCENTRAL                       3.89e-06 ***
## agricultural_districtCENTRAL COAST                 1.48e-05 ***
## agricultural_districtCENTRAL COASTAL               0.660365
## agricultural_districtCENTRAL PIEDMONT              0.000156 ***
## agricultural_districtCROSS TIMBERS                 0.339551
## agricultural_districtEAST                          5.21e-14 ***
## agricultural_districtEAST CENTRAL                  1.21e-05 ***
## agricultural_districtEASTERN                       0.000418 ***
## agricultural_districtEASTERN OR MOUNTAIN           0.634010
## agricultural_districtEDWARDS PLATEAU               0.312091
## agricultural_districtLOWER EASTERN SHORE           0.000101 ***
## agricultural_districtMIDWESTERN                    2.51e-05 ***
## agricultural_districtNORTH                         2.07e-07 ***
## agricultural_districtNORTH CENTRAL                 1.40e-08 ***
## agricultural_districtNORTHEAST                     3.60e-06 ***
## agricultural_districtNORTHERN                      0.000179 ***
## agricultural_districtNORTHERN COAST                0.448776
## agricultural_districtNORTHERN COASTAL              0.864098
## agricultural_districtNORTHERN HIGH PLAINS          0.254721
## agricultural_districtNORTHERN LOW PLAINS           0.220123
## agricultural_districtNORTHERN MOUNTAIN             0.035856 *
## agricultural_districtNORTHERN PIEDMONT             0.044497 *
## agricultural_districtNORTHWEST                     1.66e-07 ***
## agricultural_districtNORTHWEST AND MOUNTAIN        0.868323
## agricultural_districtNORTHWESTERN                  0.392780
## agricultural_districtOTHER DISTRICTS, ALL COUNTIES 0.281030
## agricultural_districtPANHANDLE                     0.675297
## agricultural_districtSACRAMENTO VALLEY             0.000518 ***
## agricultural_districtSAN JOAQUIN VALLEY            1.43e-08 ***
## agricultural_districtSAN LUIS VALLEY               3.45e-13 ***
## agricultural_districtSIERRA MOUNTAINS              0.151561
## agricultural_districtSISKIYOU-SHASTA               6.31e-05 ***
## agricultural_districtSOUTH                         0.461892
## agricultural_districtSOUTH CENTRAL                 2.34e-06 ***
## agricultural_districtSOUTHEAST                     5.58e-05 ***
## agricultural_districtSOUTHEASTERN                  2.60e-05 ***
## agricultural_districtSOUTHERN                      0.001913 **
## agricultural_districtSOUTHERN CALIFORNIA           0.052157 .
## agricultural_districtSOUTHERN COASTAL              0.409959
## agricultural_districtSOUTHERN LOW PLAINS           0.303617
## agricultural_districtSOUTHERN PIEDMONT             0.001933 **
## agricultural_districtSOUTHWEST                     0.002015 **
## agricultural_districtSOUTHWESTERN                  0.111892
## agricultural_districtTRANS-PECOS                   0.616773
## agricultural_districtUPPER EASTERN SHORE           4.20e-09 ***
## agricultural_districtUPPER PENINSULA               0.001466 **
## agricultural_districtWEST                          0.000733 ***
## agricultural_districtWEST CENTRAL                  0.000272 ***
## agricultural_districtWESTERN                       0.045042 *
## agricultural_districtWESTERN MOUNTAIN              0.310232
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.694 on 3352 degrees of freedom
## Multiple R-squared:  0.2764, Adjusted R-squared:  0.2651
## F-statistic: 24.62 on 52 and 3352 DF,  p-value: < 2.2e-16
```

- Price is an endogenous variable in our model. Provide examples of two different types of endogeneity that could bias our estimated coefficient on price.

Many correlated missing regressor. I.e Farmers storage facility, weather and “time of harvest(farming)”

- We can somewhat mitigate this problem by including year and state fixed effects. Run this regression on the provided data and report the estimated coefficient on price, along with its standard error. Justify the method you used to adjust the standard error. Is this coefficient causally interpretative?

Yes, holding the two fixed, the log(price coefficient changed and obliviously the adjusted squared increase)

log(price__per__bushel) 0.143390 (se:0.093462) Adjusted R-squared: 0.554 (55.4%)

```
reg<-lm(log(avarege_production_in_bushel)~log(price_per_bushel) + year + state, dat=agricultural_distric
summary(reg)
```

```
##
## Call:
## lm(formula = log(avarege_production_in_bushel) ~ log(price_per_bushel) +
##     year + state, data = agricultural_district_mean_production)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8588 -0.9012 -0.0068  0.9463  4.5063
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            82.656018   8.470432   9.758  < 2e-16 ***
## log(price_per_bushel)   0.143390   0.093462   1.534 0.125072
## year                   -0.034790   0.004269  -8.150 5.08e-16 ***
## stateCALIFORNIA        -0.010885   0.199037  -0.055 0.956391
## stateCOLORADO          -0.157926   0.207265  -0.762 0.446142
## stateDELAWARE          -0.146645   0.250069  -0.586 0.557634
## stateIDAHO              2.639046   0.211597  12.472  < 2e-16 ***
## stateKANSAS            -2.008283   0.239288  -8.393  < 2e-16 ***
## stateKENTUCKY          -1.041364   0.261229  -3.986 6.85e-05 ***
## stateMARYLAND          -0.296933   0.212523  -1.397 0.162452
## stateMICHIGAN          -1.913129   0.206285  -9.274  < 2e-16 ***
## stateMINNESOTA         -0.394719   0.194716  -2.027 0.042725 *
## stateMONTANA            1.958929   0.196012   9.994  < 2e-16 ***
## stateNEBRASKA          -2.824462   0.227082 -12.438  < 2e-16 ***
## stateNEVADA            -1.936198   0.262916  -7.364 2.23e-13 ***
## stateNEW JERSEY        -2.052819   0.261587  -7.848 5.65e-15 ***
## stateNEW YORK          -2.188435   0.241954  -9.045  < 2e-16 ***
## stateNORTH CAROLINA    -1.687235   0.206735  -8.161 4.63e-16 ***
## stateNORTH DAKOTA       2.539937   0.191560  13.259  < 2e-16 ***
## stateOKLAHOMA          -3.227231   0.236849 -13.626  < 2e-16 ***
## stateOREGON            -0.110373   0.207480  -0.532 0.594782
## statePENNSYLVANIA      -0.612123   0.198138  -3.089 0.002022 **
## stateSOUTH CAROLINA    -2.906555   0.259256 -11.211  < 2e-16 ***
## stateSOUTH DAKOTA      -0.770305   0.201538  -3.822 0.000135 ***
## stateTEXAS             -2.987184   0.238029 -12.550  < 2e-16 ***
## stateUTAH               0.102387   0.215308   0.476 0.634434
## stateVIRGINIA          -0.632884   0.202259  -3.129 0.001769 **
## stateWASHINGTON         1.049863   0.206013   5.096 3.66e-07 ***
## stateWISCONSIN         -0.806772   0.204492  -3.945 8.13e-05 ***
## stateWYOMING           -0.038266   0.206764  -0.185 0.853182
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.32 on 3375 degrees of freedom
## Multiple R-squared:  0.5578, Adjusted R-squared:  0.554
## F-statistic: 146.8 on 29 and 3375 DF,  p-value: < 2.2e-16
```

- Which potential sources of price endogeneity does adding fixed effects address? Discuss the difference (if any) in results with the results above. Which sources might still remain? Make sure to provide concrete examples.

other omitted variables with their confounding effect still remain.(Weather, economic policy variable, world_c)

- To address any remaining sources of price endogeneity, suppose that we use the transportation cost between location of barley production and its market as an instrument for price. Explain why you think this is or is not a valid instrument. What challenges could arise when using this instrument in practice?

The two BEST criteria for IV are: (i) It causes variation in the treatment variable; (yes, Transportation cost causes variation in the the treatment variable, P) (ii) It does not have a direct effect on the outcome variable, only indirectly through the treatment variable (here, transportation cost causes variation directly to both variables, thus, not be a good IV)

- Suppose one of the other research assistants accidentally deletes 10% of the observations of the barley production variable. How would you expect dropping these observations to change the estimated coefficient on price and its standard error if the deletions are at random? What if the deletions are not at random?

If the deletion random: we can rework on the 90% remaining data (with cross validation) and then we can check what happen with the coefficient, it all depends(we can tell after recheck)

If the deletion is not random: it is complicated and hard to tell and simply, I don't know.